ES ELECTRONIC SYSTEMS

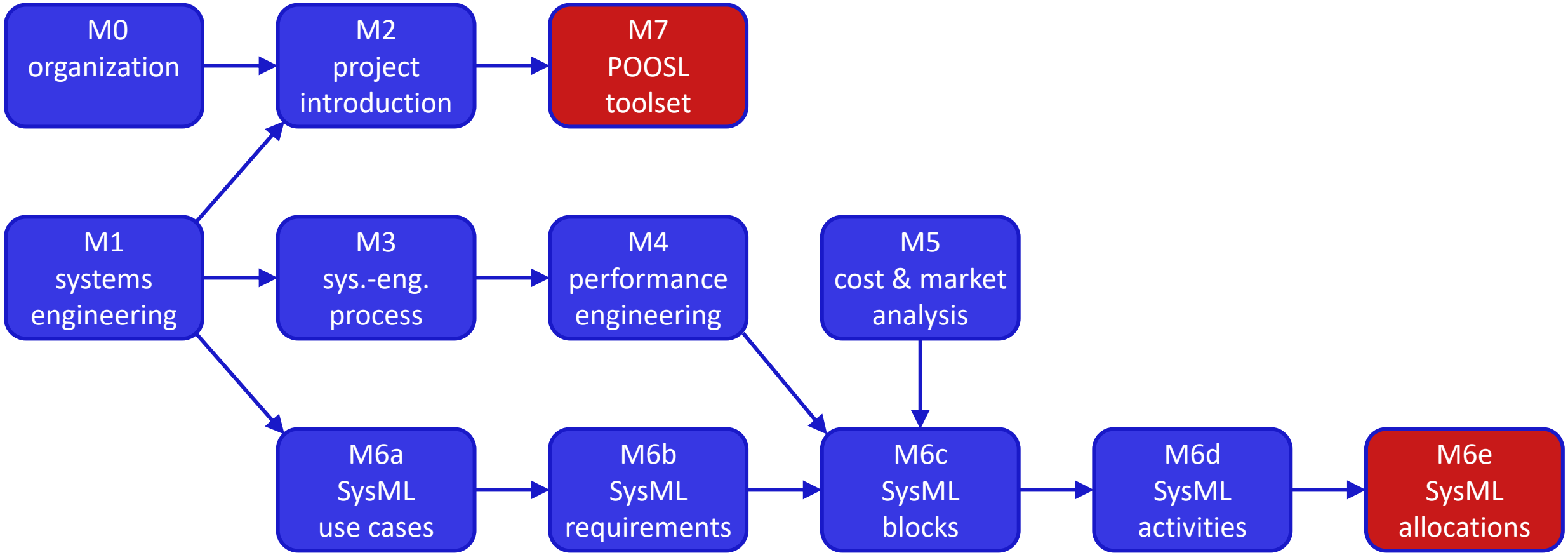
TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

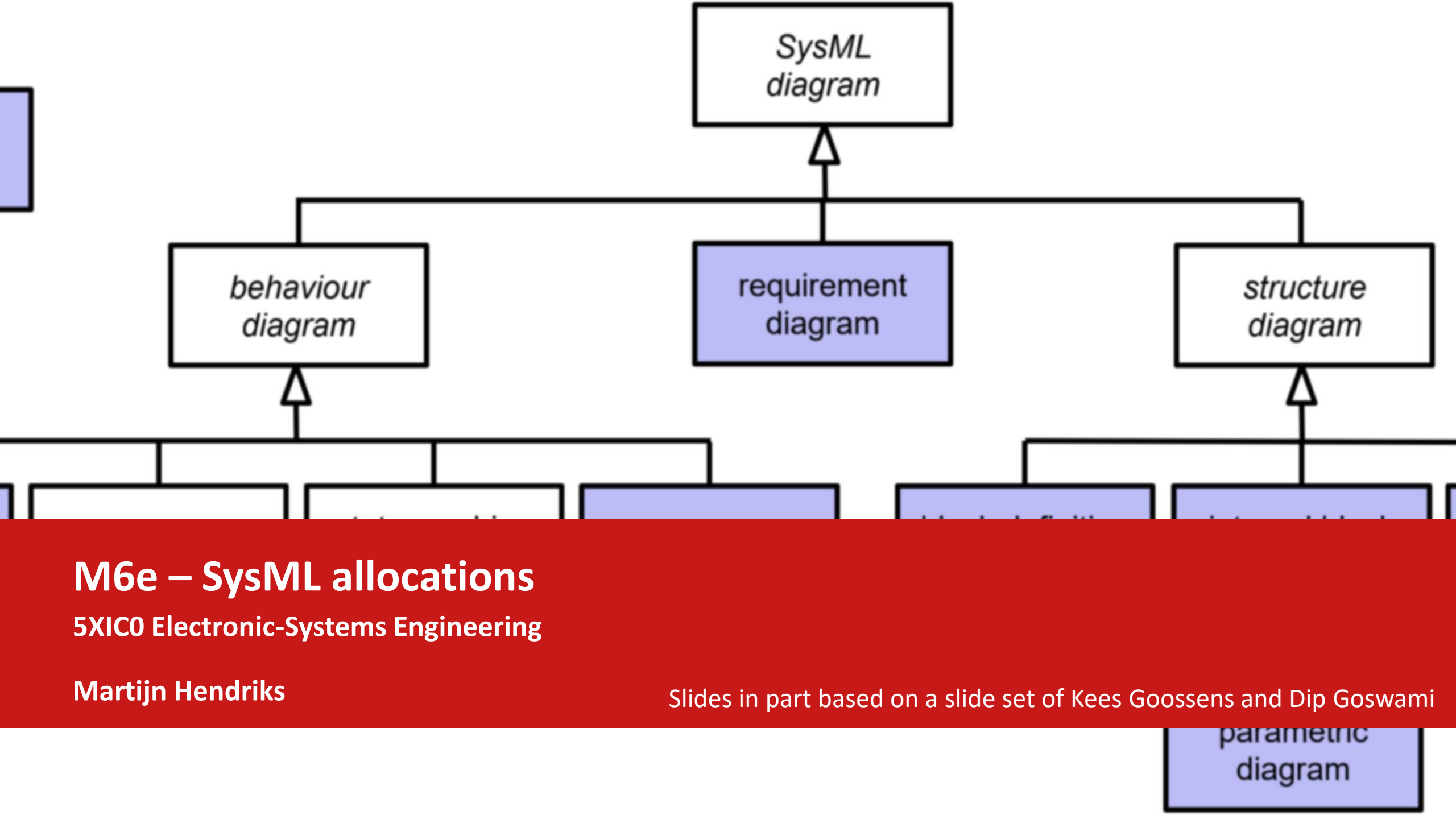
5XIC0 Electronic-Systems Engineering

Twan Basten, Martijn Hendriks

Electrical Engineering

modules


M0
organization
M2
project
introduction
M7
POOSL
toolset
M1
systems
engineering
M3
sys.-eng.
process
M4
performance
engineering
M5
cost & market
analysis
M6a
SysML
use cases
M6b
SysML
requirements
M6c
SysML
blocks
M6d
SysML
activities
M6e
SysML
allocations



M6e – SysML allocations

5XIC0 Electronic-Systems Engineering

Martijn Hendriks

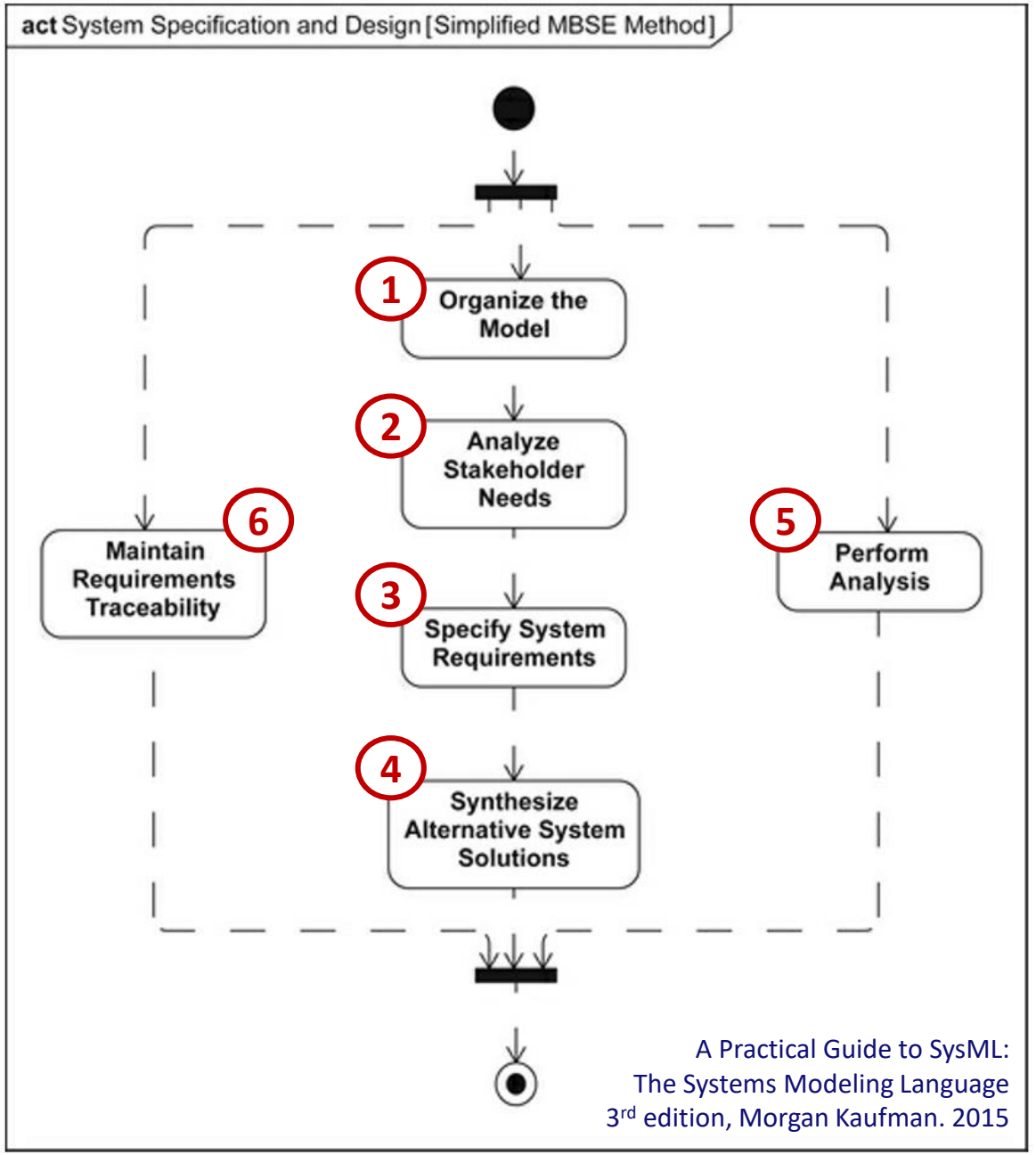
Slides in part based on a slide set of Kees Goossens and Dip Goswami

in this lecture

SysML allocations

a simplified^2 MBSE method

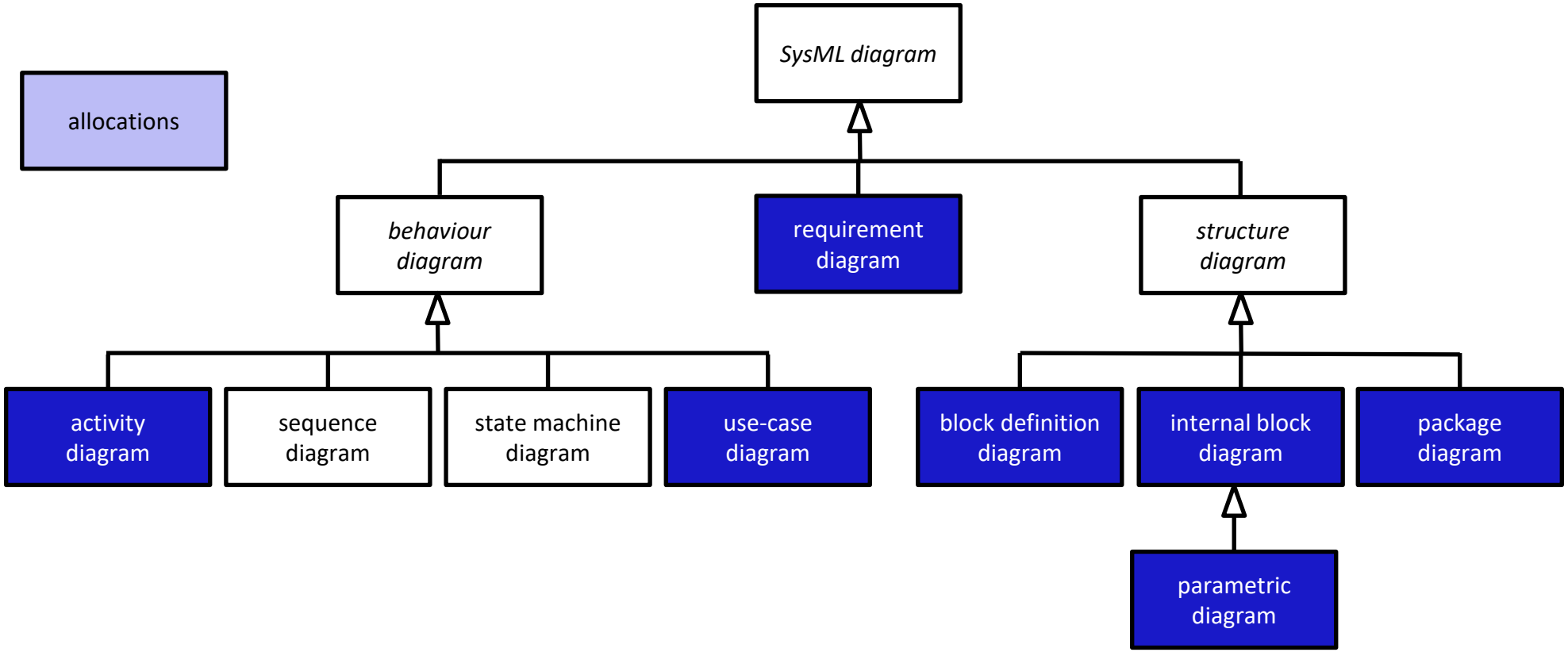
1. SysML package diagram
2. stakeholders
   SysML UC diagrams, UC descriptions
   measures of effectiveness (moes)
3. SysML requirement diagrams
4. create multiple alternatives
   - SysML BDDs – system decomposition
   - SysML IBDs – interconnections
   - SysML Activity diagrams – UC refinements
   - **SysML Allocations – activities to blocks**
5. - SysML PAR diagrams – covering all moes
   - POOSL models – makespan
   - analytical model – profit
   - verification
6. **- SysML Allocation – reqs to blocks/activities**



A Practical Guide to SysML:
The Systems Modeling Language
3rd edition, Morgan Kaufman. 2015

SysML – diagram overview

diagrams are **views** on the model
(i.e., on a subset of model elements)



SysML – allocations

a system usually has multiple descriptions

- structural / logical structural
  - defining the decomposition in its parts / usages / components
  - bdd, ibd, par
- functional / behavioural
  - defining the decomposition in its sub-functions / sub-behaviours
  - uc, act

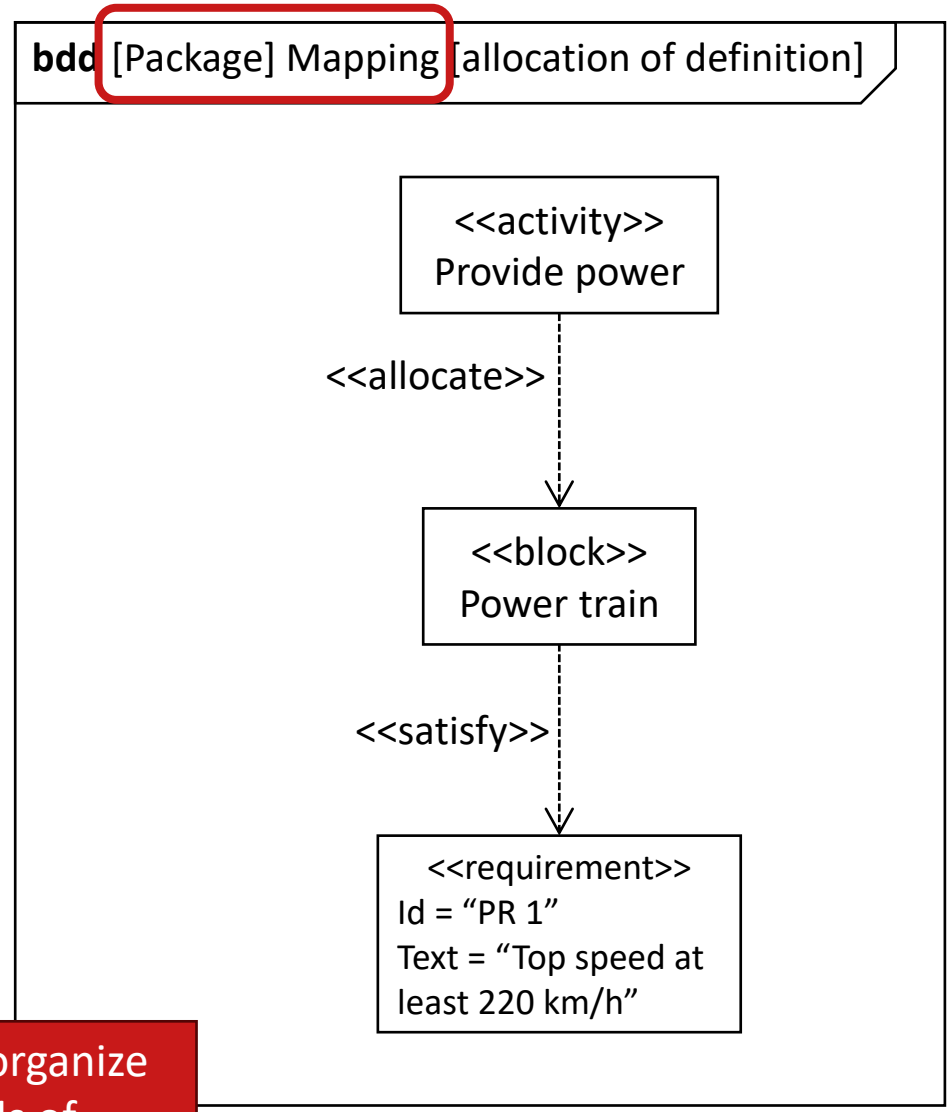
SysML – allocations

allocations define how different descriptions are related

- how are requirements realized by the design
- how is behavior (i.e., use cases and refinements with activities) implemented
- how is a logical architecture implemented by a physical architecture
- …

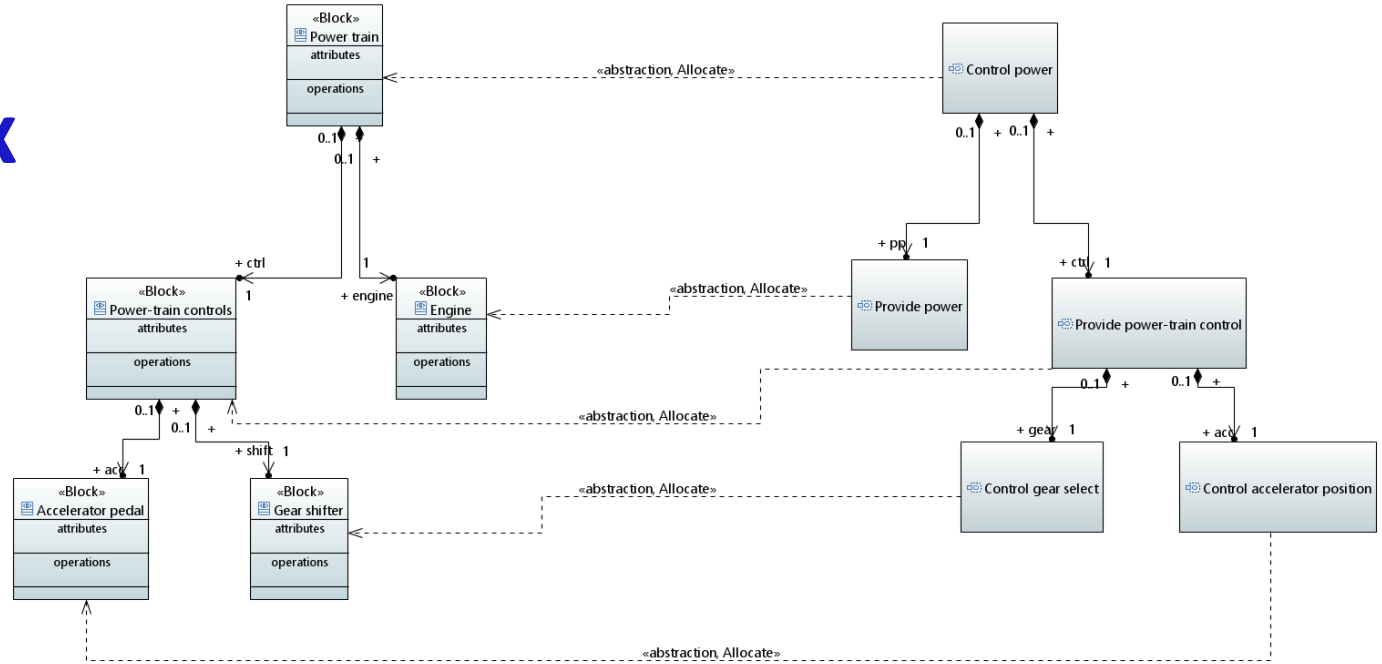
there is no special diagram (because of cross-cutting nature)

- visualize them in (separate) diagrams; bdd / req
- record them in a table/matrix



In Papyrus, you can organize the different kinds of allocations also in packages

SysML – allocation matrix

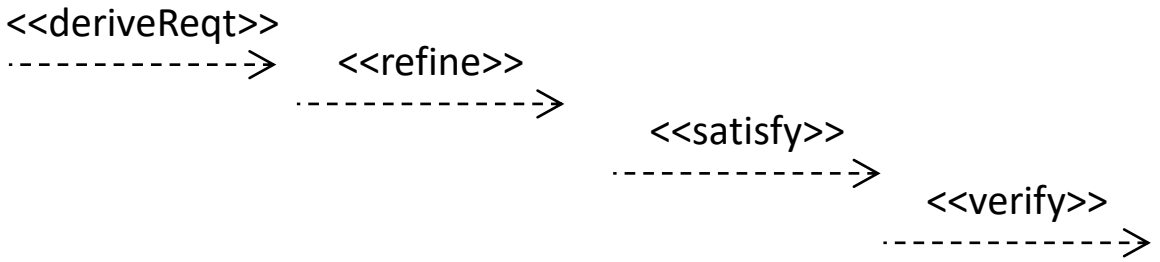


Activity / Block	Power train	Engine	Power-train controls	Accelerator pedel	Gear shifter
Control power	X				
Provide power		X			
Provide power-train ctrl			X		
Control gear select					X
Control accelerator pos				X	

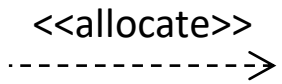
SysML – allocations

we look at three kinds of allocations (all relationships between model elements)

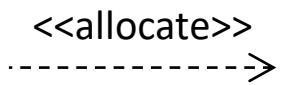
- requirement allocation
  - deriveReqt relationship
  - refine relationship
  - satisfy relationship
  - verify relationship



- functional allocation
  - allocation relationship



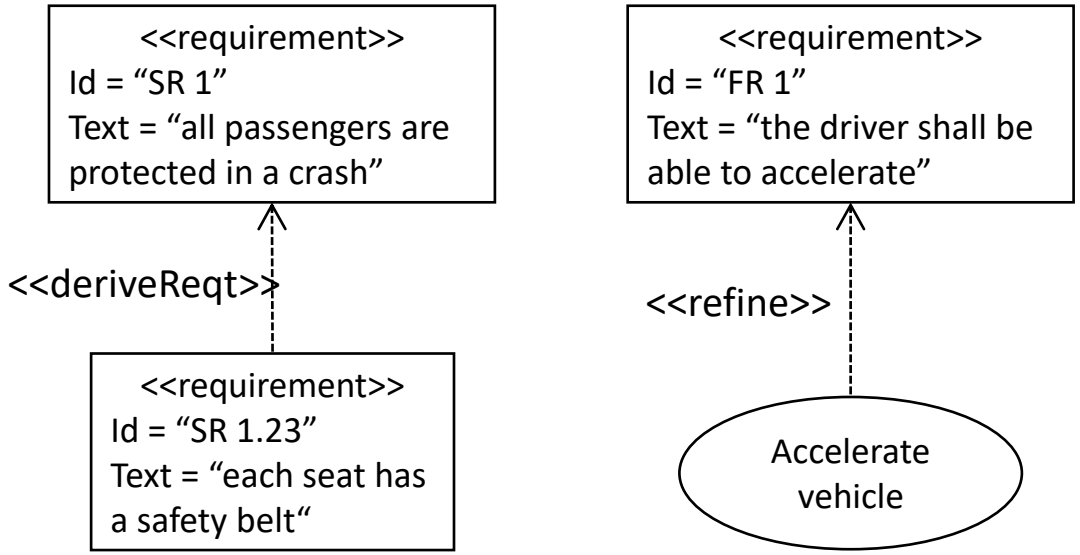
- structural allocation
  - allocation relationship



SysML – requirement allocation

- deriveReqt : requirement to requirement; states that the src requirement is derived from the dst requirement

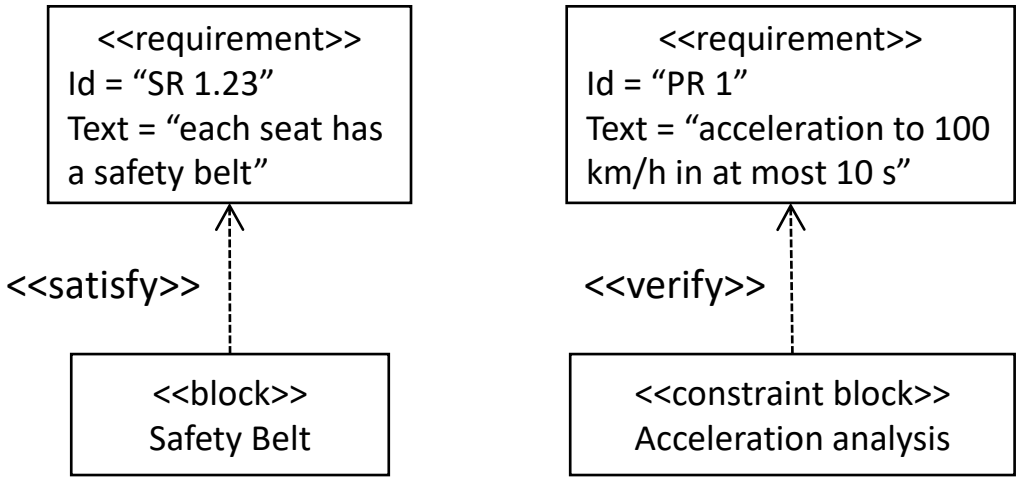
- refine : between a model element and a requirement (can be both ways); reduces ambiguity, clarifies



SysML – requirement allocation

- satisfy : model element (block, activity) to requirement; asserts that the requirement is satisfied by the model element

- verify : constraint block to requirement; proves that the requirement is satisfied by the analysis specified in the constraint block



SysML – functional allocation

decouples form from function: Y-chart pattern

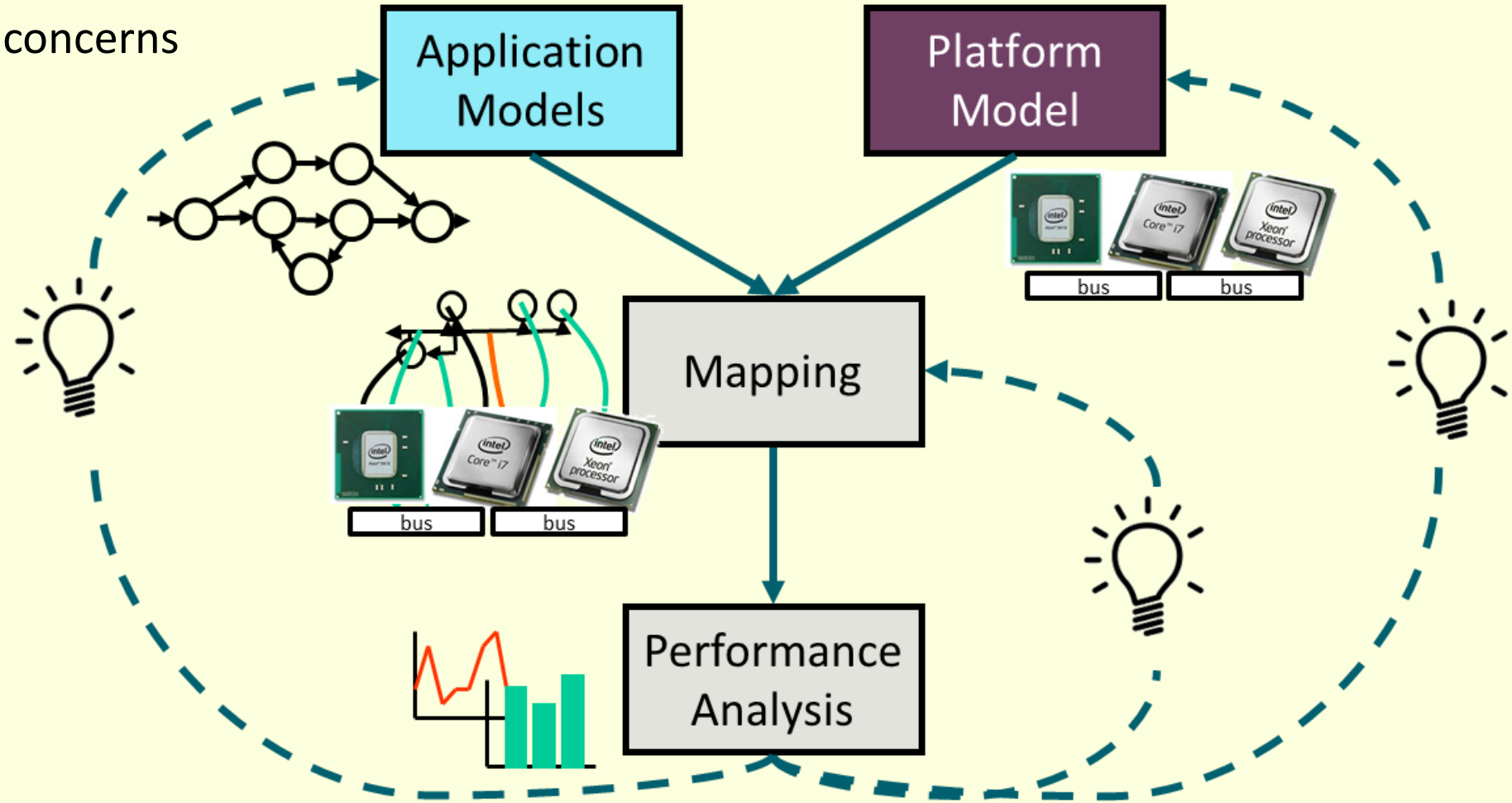
- application: behavior: use cases, activities
- platform: structure: logical and physical: blocks
- mapping of application to platform: relation between form and function

advantages:

- facilitates re-use
- relative independent development of behavior and structure
- provides an efficient way of generating multiple designs for trade off studies

Y-chart

- separation of concerns
- model reuse



source: Kienhuis et al. ASAP 1997

xCPS – new system

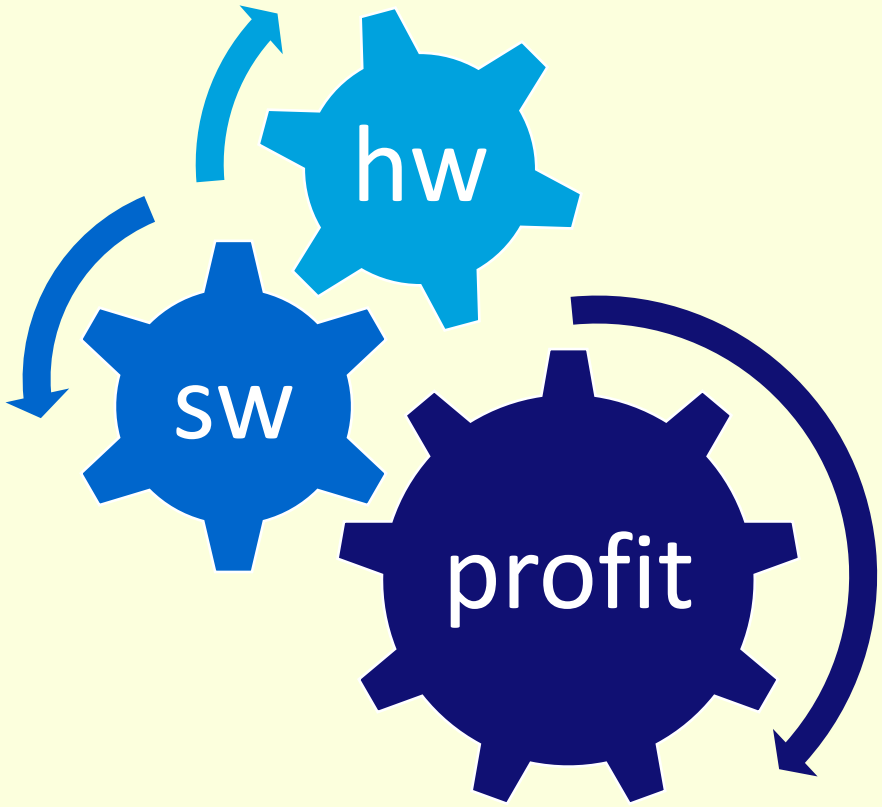
variation points

- hardware
  - slow, normal or fast gantry arm(s)
  - slow, normal or fast index table
  - slow, normal or fast belts
- software
  - piece logistics

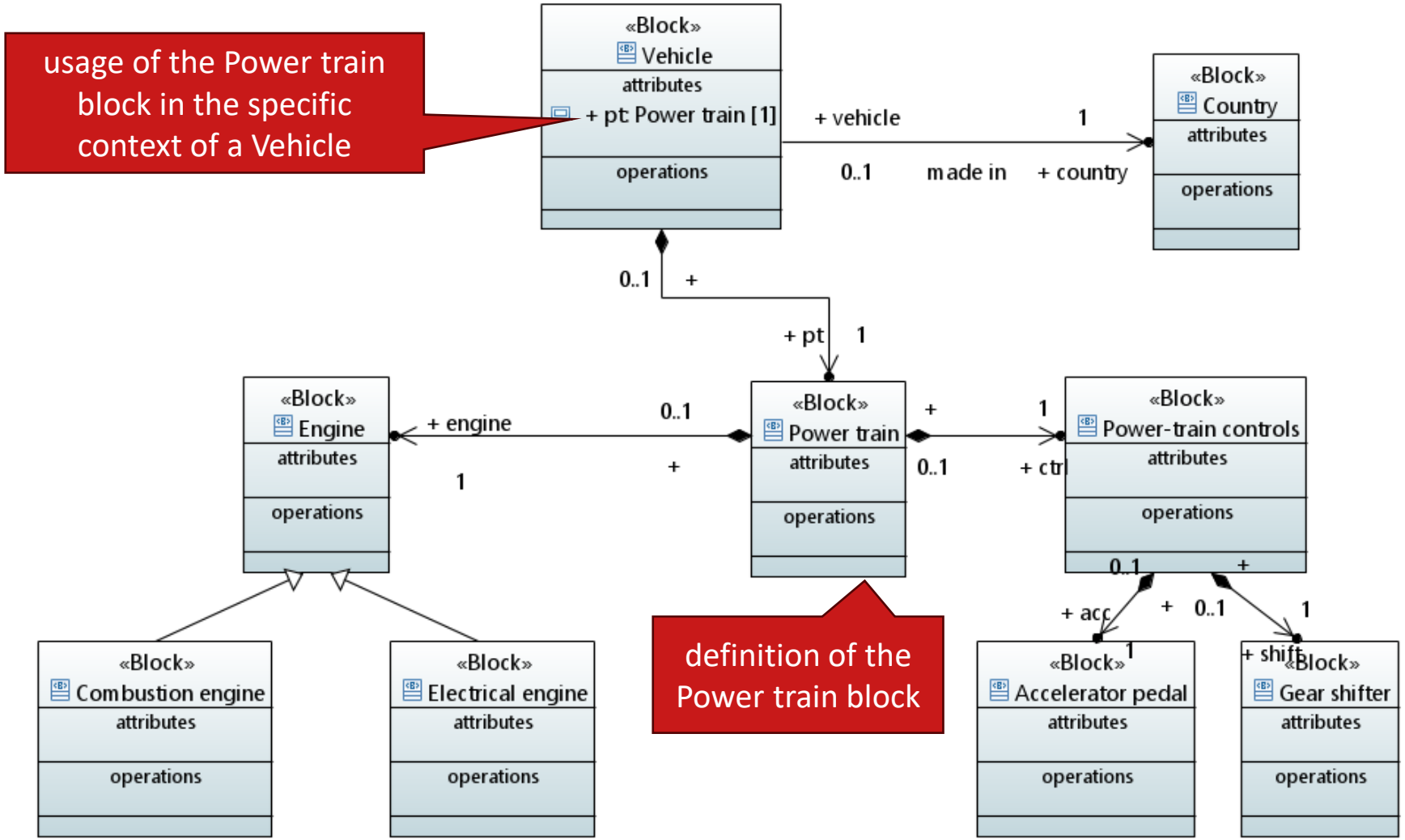
variation impacts

- batch makespan
- bill-of-material
- engineering cost
- time-to-market
- risk

…and all this impacts the profit



SysML – functional allocation – definition vs usage



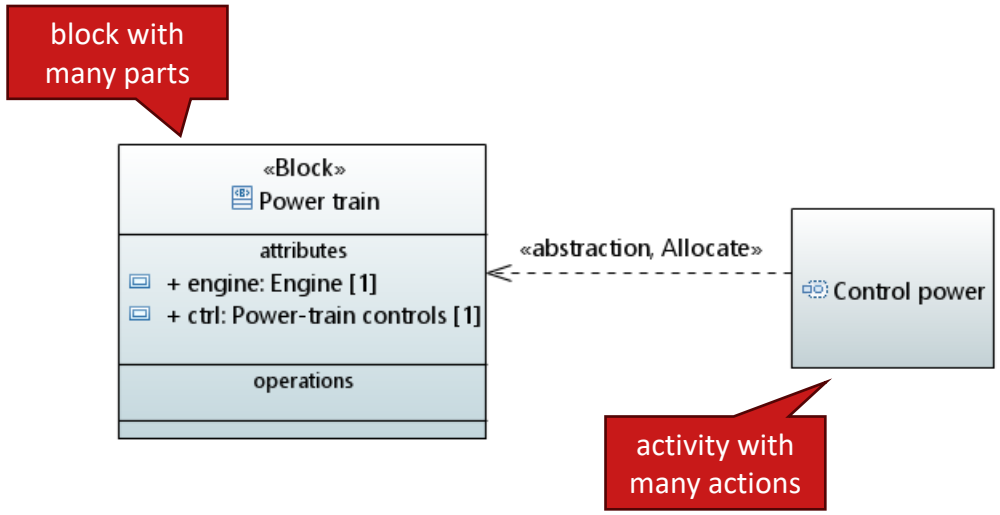
SysML – functional allocation – definition vs usage

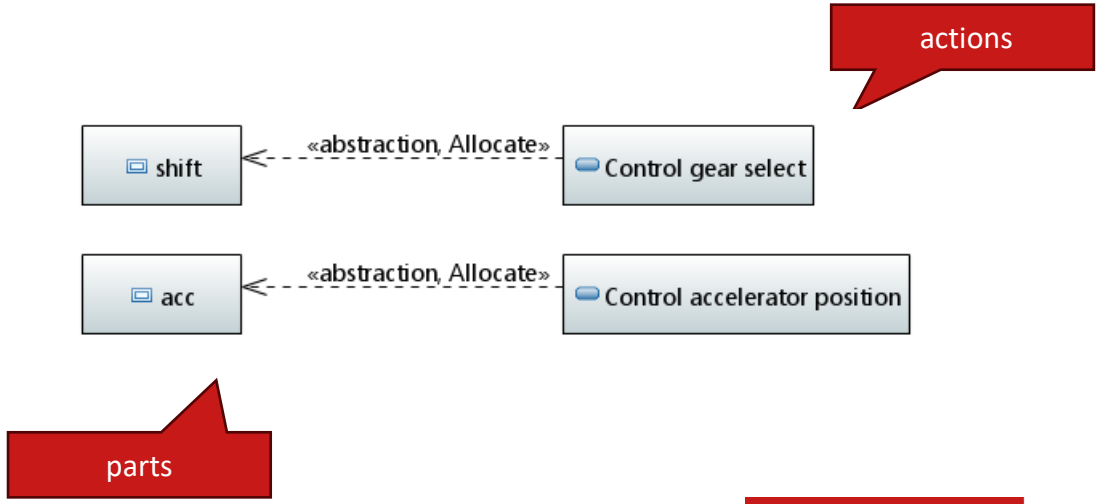
allocation of definition: activity to block

- when the allocation is intended to apply to all usages
- can result in over-allocation, i.e., more actions allocated to a part than necessary

allocation of usage: action to part

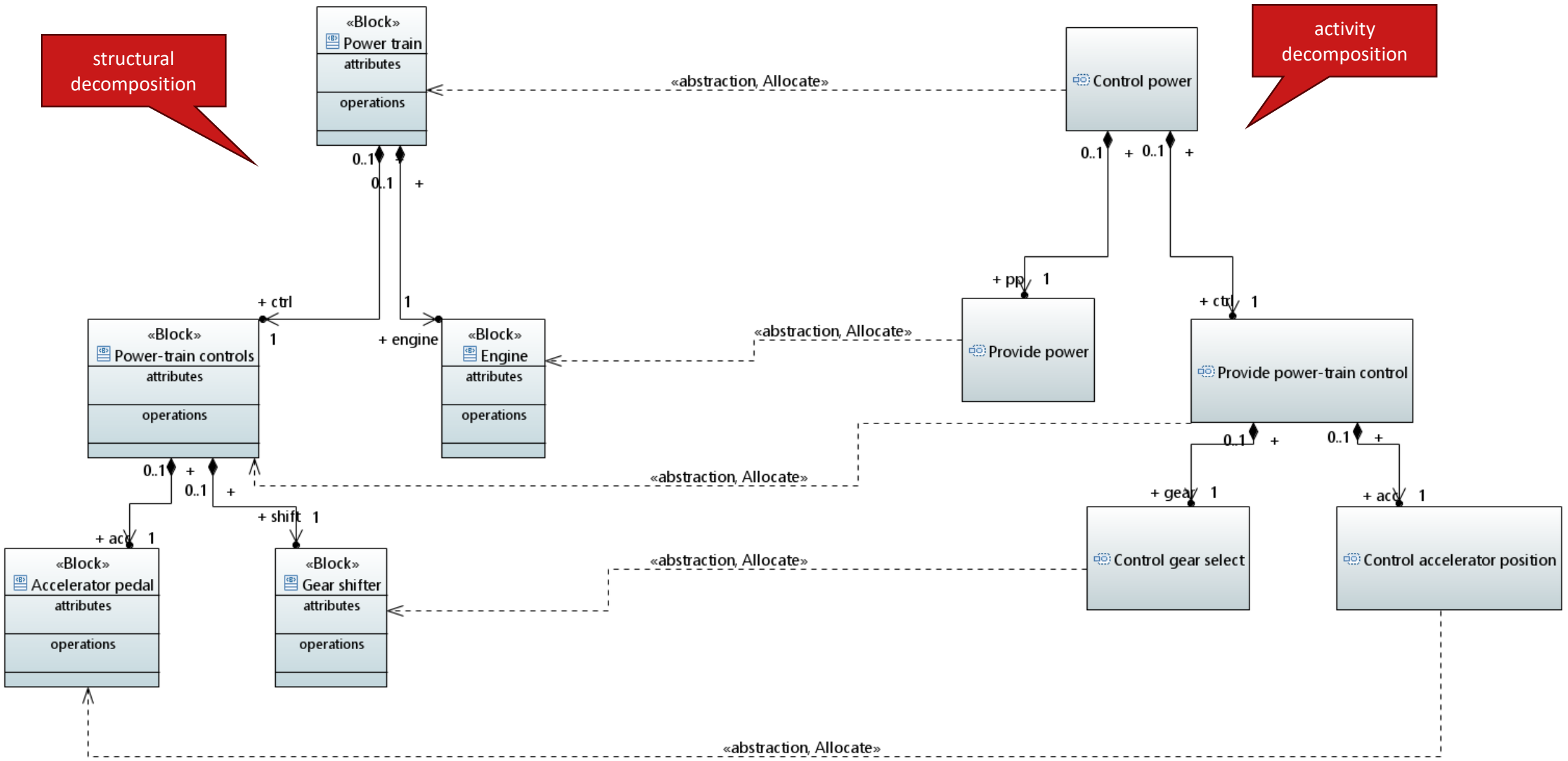
- when the allocation is not intended to be re-used
- possible redundancy or inconsistency because parts/actions can be used in multiple places





SysML – functional allocation – definition vs usage

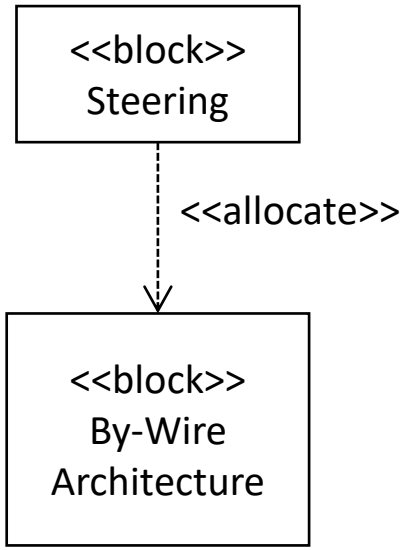
- start with allocation of usage;
- examine each of the uses, then consider allocation of definition
- allocation of definition requires blocks/activities to be specialized or decomposed to the point where the allocation of definition is unique, and over-allocation is avoided


structural decomposition
activity decomposition
«Block»
Power train
attributes
operations
«abstraction, Allocate»
Control power
0..1
+
0..1
+
+ ctrl
1
+ engine
1
«Block»
Power-train controls
attributes
operations
«Block»
Engine
attributes
operations
«abstraction, Allocate»
+ pp
1
Provide power
+ ctrl
1
Provide power-train control
«abstraction, Allocate»
0..1
+
0..1
+
+ shift
1
+ gear
1
+ acc
1
«Block»
Accelerator pedal
attributes
operations
«Block»
Gear shifter
attributes
operations
«abstraction, Allocate»
Control gear select
Control accelerator position
«abstraction, Allocate»

SysML – structural allocation

multiple logical architectures to model the what

- physical parts
- electrical / power supply architecture
- network architecture
- software architecture
- etc.



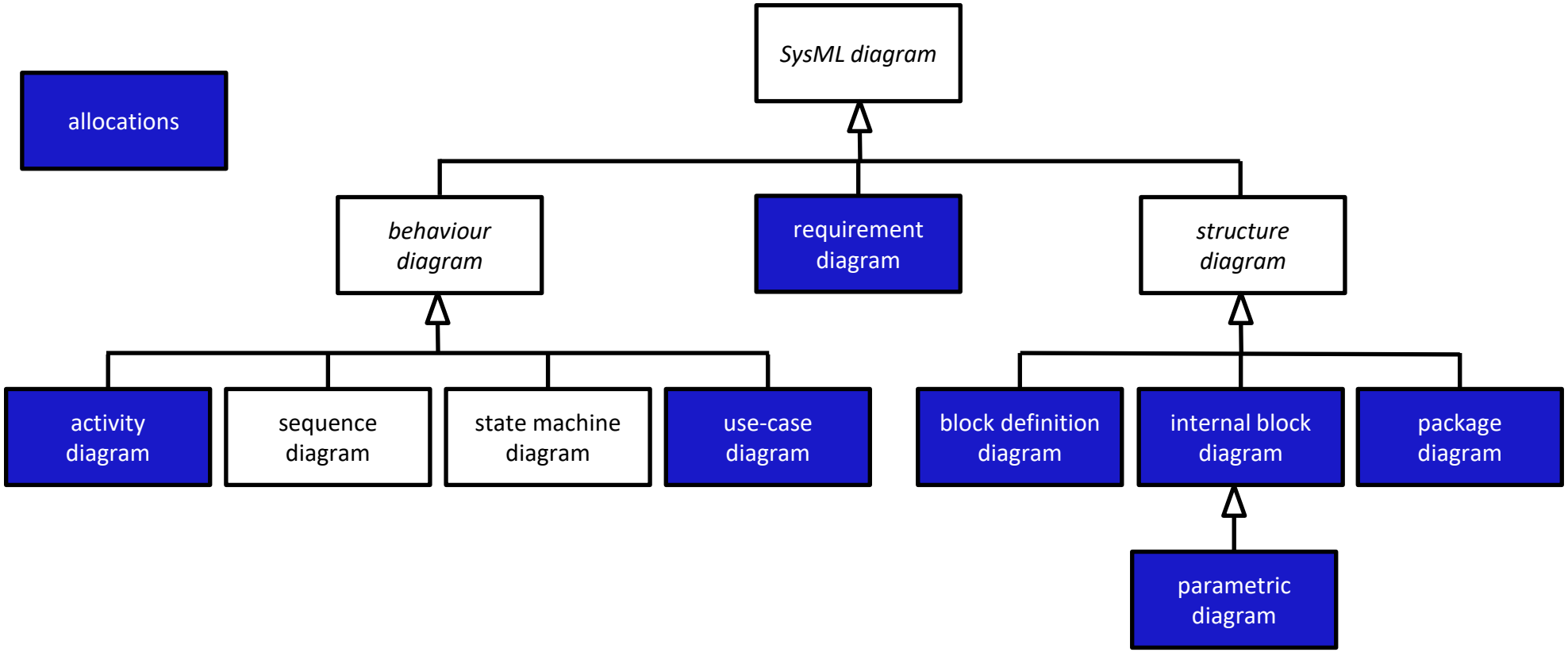
in the end each logical component will be implemented by a physical part: the how

SysML allocations – recommended reading

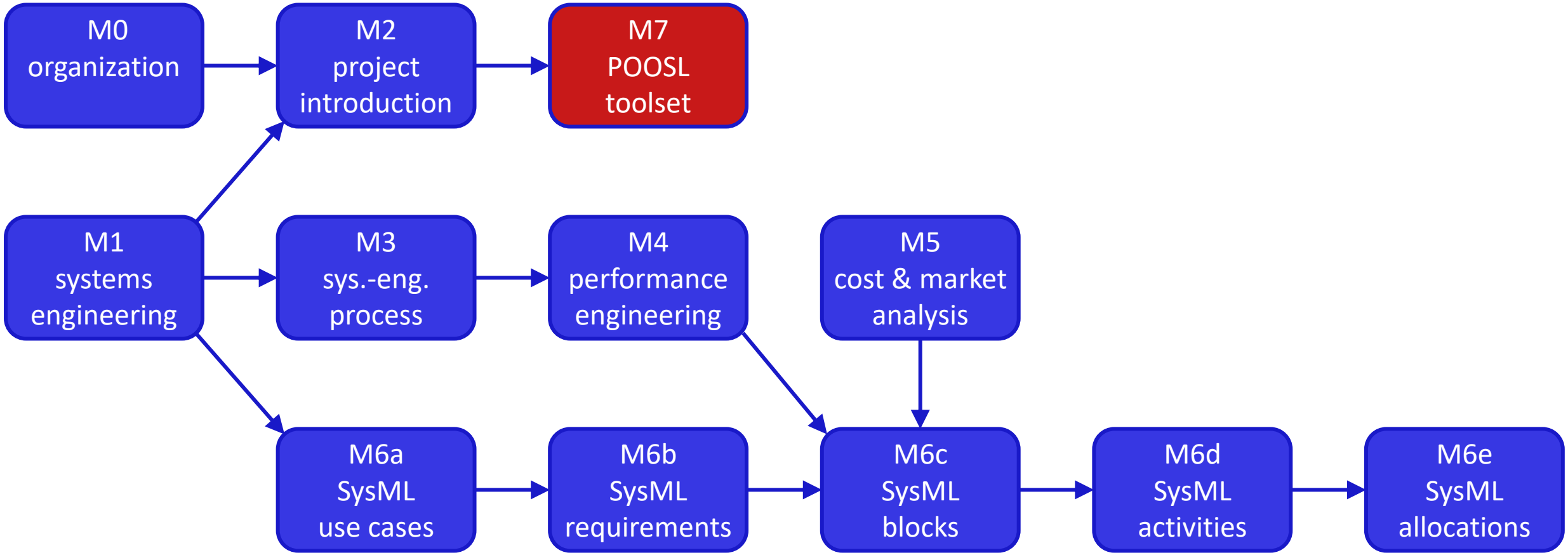
- 13.4, 13.5.1, 13.10 – 13.12
- 14.1, 14.2, 14.3, 14.4.1, 14.4.2, 14.4.4, 14.4.6, 14.5, 14.6.1, 14.6.2

SysML – diagram overview

diagrams are **views** on the model
(i.e., on a subset of model elements)



modules


M0
organization
M2
project
introduction
M7
POOSL
toolset
M1
systems
engineering
M3
sys.-eng.
process
M4
performance
engineering
M5
cost & market
analysis
M6a
SysML
use cases
M6b
SysML
requirements
M6c
SysML
blocks
M6d
SysML
activities
M6e
SysML
allocations

to remember

allocations define how different descriptions (structural/logical/functional) are related

- requirement allocation
- functional allocation
- structural allocation

there is no special allocation diagram; REQ or BDD can be used used, or matrix/table notation

functional allocation is used to realize the Y-chart modeling pattern

note

- today: deadline for feedback on your current Papyrus model
- next week: midterm exam
- after that: last module on POOSL => install POOSL and TRACE4CPS